REMARKS

This Amendment is in response to the Office Action dated September 14, 2007. In the Office Action, claim 25 was objected to, claims 1-35 were rejected under 35 U.S.C. § 112, claims 1, 2, 3-6, 11, 12, 15-19, 22, 23, 25-30 and 34 were rejected under 35 U.S.C. § 102, and claims 7-10, 13, 14, 20, 21, 24, 31-33 and 35 were rejected under 35 U.S.C. § 103. By this Amendment, claims 1-3, 5, 13-17, 19, 25-27 and 29 are amended. Currently pending claims 1-35 are believed allowable, with claims 1, 14, 15 and 25 being independent claims. In support of claim allowance, the Applicants submit the following:

CLAIM OBJECTIONS

Claim 25 stands objected to, apparently because of the use of the word "coupled". No reason was provided by the Examiner as to why claim 25 as written is no improper form.

Coupled is defined as "1 a: to connect for consideration together b: to join for combined effect 2 a:to fasten together : LINK b: to bring (two electrical circuits) into such a close as to permit mutual influence." Webster's Ninth New Collegiate Dictionary, Merriam-Webster Inc. (1989). The Applicants respectfully submit the term "coupled" is clearly used appropriately in the claim, and that claim 25 is in proper form. Thus, the objection is traversed.

REJECTIONS UNDER 35 USC §112

Claims 1, 15 and 15 were rejected under 35 U.S.C. § 112, second paragraph, as lacking antecedent basis for the term "the work load". By this amendment, claims 1, 15 and 25 are amended to provide antecedent basis for the term "the work load".

Claims 1, 2, 4-16, 18-26, and 28-35 stand rejected under 35 U.S.C. § 112, second paragraph, as allegedly "unclear and indefinite." OA, section 4B, pages 2-11. The applicants respectfully submit that requirements of 35 U.S.C. § 112, second paragraph, are misapplied in the Office Action.

According to 35 U.S.C. § 112, second paragraph, "The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention." The purpose of claims is not to explain the technology or how it works, but to state the legal boundaries of the patent grant. A claim is not

"indefinite" simply because it is hard to understand when viewed without benefit of the specification. S3 Inc. v. nVIDIA Corp., 259 F.3d 1364, 59 USPQ2d 1745 (Fed. Cir. 2001) citing Autogiro Co. of America v. United States, 384 F.2d. 391, 397, 155 USPQ 697, 701 (Ct. Cl. 1967). "The test for definiteness under 35 U.S.C. 112, second paragraph, is whether 'those skilled in the art would understand what is claimed when the claim is read in light of the specification.'" M.P.E.P. 2173.02 quoting Orthokinetics, Inc. v. Safety Travel Chairs, Inc., 806 F.2d 1565, 1576, 1 USPQ2d 1081, 1088 (Fed. Cir. 1986) (emphasis added).

In the present application, the Examiner alleges various aspects of the claims are "unclear and indefinite." More particularly, the Office Action poses a series of questions for claims 1, 2, 4-16, 18-26, and 28-35. As discussed above, however, the claims are to be interpreted in light of the specification, not in a vacuum. Thus, the rejection of claims 1, 2, 4-16, 18-26, and 28-35 under 35 U.S.C. § 112, second paragraph, is in error.

Nevertheless, in an effort to help the Examiner understand the invention, the Applicants will attempt of answer the questions posed in the Office Action below. The Examiner is reminded, however, that details about the various embodiments of the invention discussed below are not required to be recited in the claims under 35 U.S.C. § 112, second paragraph. Along with the discussion below, the Examiner is referred to specification for more details about the invention.

The present invention broadly focuses on solutions to dynamically adjust the number of servers executing an application in the event of an overload condition. It primarily does this by a novel means to segment and distribute data items across servers where:

a) Data items are kept on a single server if they are deemed "similar" using some criteria, e.g. objects are close to each other on a map

b) Servers run a distributed protocol with no centralized component to distribute data in response to "hot spots", or conditions of overload, and consolidate data when the hot spot disappears. In our invention, servers have no knowledge of other servers and run a robust and resilient protocol when distributing and consolidating data items.

c) To enable both of the above, embodiments of the invention use a "key" encoding scheme that can specify arbitrary partitioning strategies of data, and execute a distributed network protocol that assigns partitions to

servers without the use of centralized components.

Use of a key system implicitly means that the entirety of application data has similarity metrics that can be used to decompose the entirety of data into a hierarchy of parent-child relationships. In an ideal world, no partitioning is necessary and all data can be located on a single server. However, overload conditions will force that partitioning of data across multiple servers. At any one time, there are active servers (resources) executing application logic and available but otherwise idle servers that are not executing the application logic or hosting a data partition.

In an embodiment of the invention, workload of an active resource and parent workload refers to 1) the current workload of a single server that is actively executing an application and 2) the relationship of this server's data partition to the otherwise abstract concept of the parent-child hierarchy described by the key scheme. Thus, in claim 1, a "collection of workload units" is the same as a "collection of units", however, a collection of units refers to a generic workload, while a collection of workload units refers to a parent workload group.

In an embodiment of the invention, the depth of the key scheme refers to the depth within the abstract parent-child hierarchy that identifies the parent workload group currently assigned to the active resource. This can be specified as an integer with range 0 to the height of the tree. This depth can correspond to a prefix of the encoding scheme that uniquely identifies a parent workload group. For example, a Quad Tree is a data structure that can be used to describe partitions of a rectangle. In a Quad Tree, a rectangle can be split into 4 partitions of equal length and width, and each of these partitions can be recursively split into 4 partitions of equal length and height recursively to some arbitrary total number of partitions. If we assign a binary encoding key to this partitioning scheme, then each of the first 4 partitions can be assigned a binary number such as "00" for the first partition, "01", for the second, "10" for the third, and "11" for the fourth. Children of these partitions can be assigned the same numbers but using the binary identification of their parent as a prefix to uniquely identify them. For example, children of partition "00" would be "0000", "0001", "0010", and "0011". Note that each child has the same prefix.

In one embodiment of the invention, the prefix "00XX" refers to any key that has "00" as a prefix. This clearly describes a group of 4 possible keys and "00XX" is referred to as a group key. Thus a parent workload group is

described using the group key "00XX", and its children by their naturally assigned keys. If these children are also parents, then they also can be described a group key.

As previously stated, the depth of the key can directly correspond to the depth of the parent workload group it uniquely identifies in the parent child hierarchy. In the Quad Tree example, this corresponds to the length x 2 of the prefix of the key. If depth starts at 0, then there is only one partition represented by a group key "XXXX", which is all wildcards, assuming a parent child hierarchy of depth 2. This group key can also be used to assign a particular server to the single partition. At a depth of 1, there are now 4 distinct group key identifiers of "00XX", "01XX", "10XX", and "11XX." At depth 2, there are now 16 key identifiers. Thus, identifying child workload groups can refer to the decomposition of the parent workload group into its constituent parts as described by the key scheme.

In a particular embodiment of the invention, the key scheme is used to both 1) decompose workload groups into their constituent parts 2) assign these parts to other servers (target resources). The target resource is always different from the active resource and its workload units, parents, and key identifiers are determined by the key encoding scheme. The target resource executes any tasks or jobs that come with processing a workload group. The candidate and target resources can be the same resource.

In one embodiment of the invention, if a workload group has been partitioned into its constituent parts, then it is necessary to record that the parent key is inactive. This is because users of the application have no knowledge as to the current state of partitioning in the system and may need to locate a particular server responsible for a workload group. The user executes a distributed protocol that guesses what the depth is, and through knowledge of what keys are inactive, the system can determine the current partitioning of the workload groups and direct the user to the server currently assigned to the workload group of interest.

In an embodiment of the invention, redirecting entities operating on elements of the parent workload group to the target resource managing the child workload group is necessary. The entities may be any user (or process for that matter) that needs access to a server assigned a particular workload group. If that workload group gets partitioned, then those users must be directed to the server now responsible for the partition of interest to the user. Note, this may require the user to communicate with multiple servers.

In an embodiment of the invention, as described above, a user may have no knowledge about the current state of partitioning of data. In this case, the user sends a probe message to the system that guesses the current depth of partitioning. This probe message is sent to the server that would normally handle this "guessed" key. Upon receipt of this probe, the server determines if the "guessed" key is active (thus meaning the guess is correct). If not, the server informs the user that the guess was wrong.

In an embodiment of the invention, if the guessed key is incorrect, the client can refine its estimate by increasing the depth of the key and directing it to the system for processing. Essentially, it can keep increasing the depth of the key until it finds the correct server.

In an embodiment of the invention, the depth of the parent workload group key can be decreased in the opposite manner of increasing the key as described previously. In the previous Quad Tree example, the keys "0000", "0001", "0010", and "0011" can be collected under a group key "00XX" where XX represents a wildcard. The depth of this group key is minus 1 of the constituent keys.

In one embodiment, attributes refer to an arbitrary property of a workload unit that can be exploited to form a key scheme. For example, in the Quad tree example a rectangular area is encoded into partitions that groups neighboring partitions under the same parent. Essentially, the "location" attribute of the area is used as the key encoding scheme.

As described previously group keys (that describe partitions with children) can be distinguished from key identifiers (that describe partitions with no children, i.e, no further subpartitions). In one embodiment of the invention, a group key contains wildcards to represent all possible remaining combinations of that key. Keys with wildcards can be referred to as a subset of a key identifier because they only specify an exact prefix of a key, leaving the rest of the key unspecified. Since these keys actually describe workload group partitions, then any key also describes a potential load on the server that is assigned that workload group. Thus, key creation should be done sensitive to the potential load consequences on the server.

In one embodiment of the invention, a mapping service is any distributed protocol that can be employed to assign a workload group to a server. For example, a Dynamic Hash Table protocol can be employed by an embodiment of the invention to map a key to an available server.

In one embodiment of the invention, partitions are created due to unpredictable hot spots. Thus, the set of all active keys (recall that when a workload group is partitioned, the parent group key becomes inactive and the children keys are active) changes over time.

While it is true that changing the key identifiers does not change the workload without actually reassigning the corresponding workload groups to different servers. However, in an embodiment of the present invention, changing a key identifier automatically reassigns workgroups to different servers.

In one embodiment of the invention, an overall set of resources refers to available servers in the system at any point in time. The set of active resources refers to the servers currently assigned a workload group. Furthermore, the active resources can be a dynamically varying set of distributed resources.

The group keys that describe workload partitioning may be sent to a mapping service to reassign workload groups to servers. However, it is likely that the group keys with wild cards cannot be directly processed by the mapping service (for example, a Dynamic Hash Table protocol disallows wildcards in keys). Thus, in one embodiment of the invention, a virtual key is created with no wildcards from a group key with wildcards that the mapping service can use. For example, we can replace wildcards with real values (e.g., in the Quad Tree we can replace a wild card with a "0"). An "external service" can refer to such a mapping service.

In one embodiment of the invention, a virtual key may include a hash value. Here, the key undergoes two distinct transformations using two different hashing functions. The first hashing function maps a group key (with wildcards) to hash value A. Hash value A is then used as input to a second hashing function, which produces hash value B. Hash value A is a mapping of a group key to a key that is usable by a mapping service such as a dynamic hash table. Hash value B is used by the mapping service to lookup a server address.

<u>REJECTIONS UNDER 35 USC §102</u>

Claims 1, 2, 3-6, 11, 12, 15-19, 22, 23, 25-30 and 34 were rejected under 35 USC §102(e) as allegedly anticipated by U.S. Patent Document No. 2002/0194173 (Bjornson).

It is well settled that the examiner has the burden of making out a *prima facie* case of anticipation in the first instance by pointing out where each and every element of the claimed invention, arranged as required by the claim, is described identically in the reference, either expressly or under the principles of inherency. *See generally, In re Spada*, 911 F.2d 705, 708, 15 USPQ2d 1655, 1657 (Fed. Cir. 1990); *In re King*, 801 F.2d 1324, 1326, 231 USPQ 136, 138 (Fed. Cir. 1986); *Lindemann Maschinenfabrik GMBH v. American Hoist and Derrick Co.,* 730 F.2d 1452, 1458, 221 USPQ 481, 485 (Fed. Cir. 1984).

Before examining the claims in detail, the Applicants first submit that Bjornson is fundamentally different from the claimed subject matter of the present invention. Bjornson primarily describes performing sequence analysis (of nucleic acid sequence records of amino acid sequence records) using a cluster of computers. Bjornson, paragraph [0051]. These records may be distributed across many databases. The primary task of Bjornson is pattern matching a set of query sequences against these sequence records. Id. A centralized shared memory (called a VSM) is used to coordinate computer activities. Bjornson, paragraph [0059]. Furthermore, computers a priori choose a subsequence of the processing task based on user-defined thresholds of resources and time. Bjornson, paragraph [0054]. These subsequences are easily created by partitioning the records based on the user-defined thresholds. Bjornson, paragraph [0074].

Claim 1

Claim 1 recites, in part, "A method for dynamically adjusting a workload of an active resource, the workload being expressed as a collection of units, each unit including its own key identifier, the active resource being associated with at least one parent workload group, the parent workload group including a collection of workload units such that workload units belonging to the parent workload group share an identical sequence of values at a specified depth value of their key identifiers, the identical sequence of values defining a group key identifier associated with the parent workload group." In rejecting claim 1, the Office Action argues Bjornson discloses such limitations in Figure 5. OA, page 12. The Applicants respectfully disagree with this position.

Bjornson illustrates in Figure 5 a standard binary tree description of partitioning a set of nucleotide and/or amino acid records into constituent parts, and later recombining them into the original partition. The partition

is created from estimates of computation resource usage and computation time. Bjornson, paragraph [0131]. There is no discussion in Bjornson, including Figure 5, of dynamically adjusting a workload of an active resource. Bjornson does not contain in teaching a workload expressed as a collection of units, each unit including its own key identifier. There is no discussion in Bjornson of an active resource associated with at least one parent workload group, the parent workload group including a collection of workload units such that workload units belonging to the parent workload group share an identical sequence of values at a specified depth value of their key identifiers, the identical sequence of values defining a group key identifier associated with the parent workload group.

Claim 1 further recites, "increasing the depth value of the parent workload group such that at least two child workload groups are identified." It is noted that, according to claim 1, a parent workload group includes a collection of workload units such that workload units belonging to the parent workload group share an identical sequence of values at a specified depth value of their key identifiers. The Office Action alleges such teaching is found at paragraphs [0060] and [0061] of Bjornson. OA, page 12. The Applicants respectfully submit such a conclusion is in error.

Paragraphs [0060] and [0061] of Bjornson state,

> [0060] Execution of a searching task requires some quantity of computational resources (e.g., memory, disk, CPU time, etc.), and upon taking a task, a worker computer estimates the quantity of computational resources required to execute the task. This estimate is termed "RES(Task)." If RES(Task) is too large for that particular worker computer, the worker computer will divide the searching task into two smaller searching tasks and add one of them to the Task List kept in the VSM bulletin board. RES(Task) will then be recalculated for the one of the two smaller searching tasks retained by the worker computer. The two smaller searching tasks that are the parts of the now-divided searching task are termed "Buddies." Each new smaller searching task is marked as the other one's Buddy, and the original undivided task is marked as the "Parent" of each of the two new smaller searching tasks.
>
> [0061] Once a worker computer obtains a task for which RES(Task) is not too large, it then estimates the fraction of the remaining computational effort represented by the task, termed GRAN(Task), and determines if it is too large. GRAN(Task) is too large if it exceeds a defined constant parameter times the ratio of the estimated computational power of the worker computer in question to the estimated total computational power of the aggregate of computers that the worker computer in question believes to be

> operating in parallel at the current time. If GRAN(Task) is too large, the worker computer goes through a similar process of dividing the searching task into two smaller searching tasks and retaining one of them as is performed when RES(Task) is too large.

It is clear from reviewing these passages, no disclosure is made of increasing the depth value of a parent workload group such that at least two child workload groups are identified. Bjornson partitioning is based on the single strategy of determining thresholds for highly predictable computation time and highly predictable server resource usage. There is no teaching in Bjornson of the recited key scheme of increasing the depth value of a parent workload group such that at least two child workload groups are identified.

For at least these reasons, it is respectfully submitted that Bjornson fails to anticipate the limitations of claim 1. Thus, claim 1 is believed allowable and indication of such allowance is earnestly requested.

<u>Claim 2</u>

Claim 2 is dependent on claim 1 and is amended to recite, "The method of claim 1, further comprising if the overload condition exists, identifying at least one candidate resource to which the child workload groups may be distributed <u>using a decentralized protocol</u>." Support for this amendment can be found at least at page 14, lines 16-27 of the present application.

As discussed above, Bjornson discloses a centralized coordination scheme for computers to coordinate their activities using a centralized shared memory called a VSM. By contrast, claim 2 recites identifying at least one candidate resource to which the child workload groups may be distributed using a decentralized protocol. Thus, claim 2 is not anticipated by Bjornson.

<u>Claim 3</u>

Claim 3 is dependent on claim 1 and is amended to recite, "The method of claim 1, further comprising requesting workload acceptance from the target resource <u>at a peer level</u>." Support for this amendment can be found at least at page 23, lines 1-10 and Figure 3 of the present application.

As discussed above, Bjornson discloses a centralized coordination scheme for computers to coordinate their activities using a centralized shared memory called a VSM. By contrast, claim 3 recites requesting workload

acceptance from the target resource at a peer level. Thus, claim 3 is not anticipated by Bjornson.

Claim 4

Claim 4 is dependent on claim 1 and recites, "The method of claim 1, further comprising recording the parent workload group as inactive at the active resource." The Office Action alleges such teaching is found at paragraph [0056] of Bjornson. OA, page 13. The Applicants strongly disagree with such a conclusion.

Paragraph [0056] of Bjornson states,

> [0056] Initially, the list of searching tasks kept in the VSM bulletin board (i.e., the Task List) contains a single task representing the entire searching task. One or more worker computers may concurrently examine the searching tasks in the Task List. During the operation of the instant method, each worker computer of the computers operating in parallel may attempt to take (i.e., copy and remove, in an atomic operation) a task from the Task List. The VSM system, such as PARADISE.RTM. for the JAVA.TM. platform, ensures that each task may be taken by at most one of the worker computers that attempt to take a task. If there is an insufficient number of tasks on the Task List to permit each worker computer attempting to take a task to take at least one, then some of the worker computers attempting to take a task from the Task List may be forced to wait either until one or more additional tasks are added to the Task List, or until a signal to exit is received.

The Applicants respectfully submit there is no disclosure in the above paragraph of recording the parent workload group as inactive at the active resource. Indeed there is no mention of parent workload group, let alone a step of recording one as inactive at an active resource.

For at least these reasons, and the reasons discussed for claim 1, it is respectfully submitted that Bjornson fails to anticipate the limitations of claim 4. Thus, claim 4 is therefore believed allowable and indication of such allowance is earnestly requested.

Claim 5

Claim 5 is dependent on claim 1 and is amended to recite, "The method of claim 1, further comprising transferring application-specific objects corresponding to the child workload groups at a peer level." Support for this amendment can be found at least at page 23, lines 1-10 and Figure 3 of the present application.

As discussed above, Bjornson discloses a centralized coordination scheme for computers to coordinate their activities using a centralized shared memory called a VSM. By contrast, claim 5 recites requesting workload acceptance from the target resource at a peer level. Thus, claim 5 is not anticipated by Bjornson.

Claim 6

Claim 6 is dependent on and further limits claim 1. Since claim 1 is believed allowable, claim 6 is also believed allowable for at least the same reasons as claim 1.

Claim 11

Claim 11 is dependent on claim 1 and recites, "The method of claim 1, further comprising associating the workload unit with the key identifier such that the key identifier encodes one or more attributes of the workload unit." The Office Action alleges such teaching is found in Figure 5 of Bjornson. OA, page 14. The Applicants strongly disagree with such a conclusion.

Bjornson describes in Figure 5 a standard binary tree description of partitioning a set of (nucleotide and/or amino acid) records into constituent parts, and later recombining them into the original partition. This fundamentally differs from claim 11, which associates a workload unit with the key identifier such that the key identifier encodes one or more attributes of the workload unit. There is no such teaching in Figure 5 of Bjornson.

For at least these reasons, and the reasons discussed for claim 1, it is respectfully submitted that Bjornson fails to anticipate the limitations of claim 11. Thus, claim 11 is therefore believed allowable and indication of such allowance is earnestly requested.

Claim 12

Claim 12 is dependent on claim 1 and recites, "The method of claim 1, further comprising constructing a virtual key for mapping to the target resource, wherein the virtual key includes a load-dependent subset of a complete key identifier." The Office Action alleges such teaching is found in Figure 5 of Bjornson. OA, page 14. The Applicants strongly disagree with such a conclusion.

Bjornson describes in Figure 5 a standard binary tree description of partitioning a set of (nucleotide and/or amino acid) records into constituent

parts, and later recombining them into the original partition. This fundamentally differs from claim 12, which constructs a virtual key for mapping to the target resource, wherein the virtual key includes a load-dependent subset of a complete key identifier. There is equivalent concept of virtual keys in Bjornson.

For at least these reasons, and the reasons discussed for claim 1, it is respectfully submitted that Bjornson fails to anticipate the limitations of claim 12. Thus, claim 12 is therefore believed allowable and indication of such allowance is earnestly requested.

Claims 15-19, 22, 23, 25-30 and 34

Claim 15-19, 22, 23, 25-30 and 34 were rejected for the same reasons as claims 1, 2, 3-6, 11 and 12. Thus, claims 15-19, 22, 23, 25-30 and 34 are believed allowable for the same reasons discussed above with regard to claims 1, 2, 3-6, 11 and 12.

REJECTIONS UNDER 35 USC §103

Claims 7-10, 13, 14, 20, 21, 24, 31-33 and 35 were rejected under 35 USC §102(e) as allegedly obvious over Bjornson.

It is well settled that "rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." In re Kahn, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336, quoted with approval in KSR Int'l Co. v. Teleflex Inc., 127 S. Ct. 1727, 1741, 82 USPQ2d 1385, 1396 (2007).

Claim 7

In rejecting claim 7, the Examiner states Bjornson "is silent as to, the specifics of receiving a probe message from an entity operating on a workload unit that is a member of the parent workload group, the probe message including a guessed identifier key formed by guessing a depth to be associated with the unit's key identifier; and sending a response to the entity indicating the group key identifier that the current resource locally determines to be the nearest known active parent group to which the element's key identifier belongs." OA, page 15. Nevertheless, the Examiner alleges, "To specifically specify the detailed steps of [claim 7] would have been obvious to one of ordinary skill in the art as it is well known that in order for the worker computer to keep track of the tasks and subtasks it is working

on, it needs to keep track of the parent groups of all of its child task groups.” The Applicants respectfully disagree with such a conclusion.

In the present invention, worker computers do not have to keep track of parent-child relationships. In fact, it is the very independence of this knowledge that allows the present invention to scale to potentially hundreds of thousands of servers, each with very limited knowledge about the actual placement of workload groups to servers. This is necessary because a server may leave the system, and the system must adapt to the absence of the server. Such an arrangement is achieved through a novel key encoding scheme and mapping service, which provides a level of indirection between servers that may hold related workload groups (e.g., same parent). The server that originally initiated a partitioning may no longer be available if the resultant sub-partitions are consolidated. Bjornson provides no facility to adapt to a dynamic set of servers.

Furthermore, the Examiner provides no documentary evidence in support of the claim 7 rejection. Official notice unsupported by documentary evidence should only be taken by an examiner when the facts asserted are well-known in the art, and are capable of instant and unquestionable demonstration as being well-known. MPEP 2144.03. If the Examiner is relying on personal knowledge to support the finding of what is known in the art, the examiner must provide an affidavit or declaration setting forth specific factual statements and explanation to support the finding. MPEP 2144.03 citing 37 CFR 1.104(d)(2).

For at least these reasons, and the reasons discussed for claim 1, it is respectfully submitted claim 7 is therefore allowable. Indication of such allowance is earnestly requested.

<u>Claim 8</u>

Claim 8 recites, “The method of claim 7, wherein the entity operating on a workload unit uses the response to further refine its estimate of a correct depth to be associated with the unit’s key identifier; and probing another resource associated with the parent key group formed by using the refined depth of the unit’s key identifier.” In rejecting claim 8, the Office Action alleges such teaching is found at paragraphs [0060] and [0061] of Bjornson. <u>OA</u>, page 16. The Applicants respectfully submit such a conclusion is in error.

Paragraphs [0060] and [0061] of Bjornson state,

> [0060] Execution of a searching task requires some quantity of computational resources (e.g., memory, disk, CPU time, etc.), and upon taking a task, a worker computer estimates the quantity of computational resources required to execute the task. This estimate is termed "RES(Task)." If RES(Task) is too large for that particular worker computer, the worker computer will divide the searching task into two smaller searching tasks and add one of them to the Task List kept in the VSM bulletin board. RES(Task) will then be recalculated for the one of the two smaller searching tasks retained by the worker computer. The two smaller searching tasks that are the parts of the now-divided searching task are termed "Buddies." Each new smaller searching task is marked as the other one's Buddy, and the original undivided task is marked as the "Parent" of each of the two new smaller searching tasks.

> [0061] Once a worker computer obtains a task for which RES(Task) is not too large, it then estimates the fraction of the remaining computational effort represented by the task, termed GRAN(Task), and determines if it is too large. GRAN(Task) is too large if it exceeds a defined constant parameter times the ratio of the estimated computational power of the worker computer in question to the estimated total computational power of the aggregate of computers that the worker computer in question believes to be operating in parallel at the current time. If GRAN(Task) is too large, the worker computer goes through a similar process of dividing the searching task into two smaller searching tasks and retaining one of them as is performed when RES(Task) is too large.

It is clear from reviewing these passages, no disclosure is made of an entity operating on a workload unit using a response to the entity indicating the group key identifier to further refine its estimate of a correct depth to be associated with the unit's key identifier. Furthermore, there is no disclosure of probing another resource associated with a parent key group formed by using the refined depth of the unit's key identifier. The claimed probe messages have no equivalent in Bjornson.

For at least these reasons, it is respectfully submitted that claim 8 is allowable.

<u>Claims 9, 20 and 32</u>

Claim 9 recites, in part, "identifying at least two workload groups for consolidation into a consolidated workload group." In rejecting claim 9, the Examiner alleges such teaching is found at paragraph [0069] and Figure 5 of Bjornson. The Applicants respectfully disagree.

Paragraph 69 and Figure 5 of Bjornson refer to the consolidation of results from queries over subsequences into their parent sequences. Such teaching is completely unrelated to identifying at least two workload groups for consolidation into a consolidated workload group, where consolidation is done to exploit excess capacity in a server that was previously fully utilized.

For at least these reasons, and the reasons discussed for claim 1, it is respectfully submitted claim 9 is allowable. Indication of such allowance is earnestly requested.

Claims 10 and 21

Claim 10 is dependent on and further limits claim 9. Since claim 9 is believed allowable, claim 10 is also believed allowable for at least the same reasons as claim 9.

Claim 13

Claim 13 recites, "The method of claim 12, further comprising using the constructed load-dependent virtual key as an input to a separate mapping service that returns the identity of the target resource to which the workload units belonging to the virtual key should be directed." In rejecting claim 13, the Examiner correctly states that Bjornson does not disclose the limitations of claim 13. The Examiner further alleges, "However, it would have been obvious to one having ordinary skill in the art at the time of the invention to let each worker computer, which corresponds to the resources, keep track of the tasks that are associated with its task ID so that it knows what part of an overall task it is currently working on, and further so that it may reconsolidate the results of each sub-tasks into one final task result in the end (Fig 5, paragraphs 63 and 69.)" OA, page 17.

As described previously, there is no equivalent to the claimed virtual key in Bjornson. In Bjornson, Figure 5 and paragraphs [0063] and [0069] refer to a binary tree that partitions sequences into subsequences. This is fundamentally different from the virtual key, which is used by a mapping service to assign a workload group to a server in a decentralized manner.

Furthermore, the Examiner provides no documentary evidence in support of the claim 13 rejection. Official notice unsupported by documentary evidence should only be taken by an examiner when the facts asserted are

well-known in the art, and are capable of instant and unquestionable demonstration as being well-known. MPEP 2144.03. If the Examiner is relying on personal knowledge to support the finding of what is known in the art, the examiner must provide an affidavit or declaration setting forth specific factual statements and explanation to support the finding. MPEP 2144.03 citing 37 CFR 1.104(d)(2).

For at least these reasons, it is respectfully submitted that claim 13 is allowable.

Claim 14

Claim 14 recites, in part, "a set of client entities utilizing the distributed application, each client entity being associated with at least one key identifier, and each client entity dynamically determining a load-dependent group of identifier keys that each such currently belongs to." In rejecting claim 14, the Office Action argues Bjornson discloses such limitations in Figure 5. OA, page 18. The Applicants respectfully disagree with this conclusion.

Figure 5 of Bjornson is fundamentally different for the above limitations in that it is applicable only to sequence analysis and require a centralized lookup mechanism (the VSM). Bjornson does not manage a set of key identifiers for the purpose of dynamically adjusting to over- and under-load conditions as these conditions never arise in the disclosure. Nor does Bjornson use a key as recited, to map a workload group to a single server through a mapping service.

For at least these reasons, it is respectfully submitted that claim 14 is allowable.

Claim 24

Claim 24 recites, "The system of claim 15, further comprising an external service configured to identify at least one candidate resource to which the child workload groups may be distributed." In rejecting claim 24, the Office Action argues Bjornson discloses such limitations at paragraph [0056]. OA, page 19. The Applicants respectfully disagree with this conclusion.

Paragraph 56 of Bjornson refers to a VSM that individual servers can use to assign themselves tasks to process. This fundamentally differs from claim 24, whereby an external service is configured to identify at least one

candidate resource to which the child workload groups may be distributed. The VSM in Bjornson is an instance of well-known shared memory architectures that are widely employed in parallel and distributed computing systems. In Bjornson, a server assigns itself a task from the VSM based on its estimated processing cost and idle/quantum time thresholds. Any server can assign itself any task. Thus, Bjornson provides no disclosure or suggestion of the limitations of claim 24.

For at least these reasons, it is respectfully submitted that claim 24 is allowable.

Claim 33

Claim 33 recites, "The computer program product of claim 31, wherein the program code to generate the consolidated key includes program code to decrease the depth value of the parent workload group such that the consolidated workload group is identified." In rejecting claim 33, the Office Action argues Bjornson discloses such limitations at Figure 5 and paragraph [0069]. OA, page 19. The Applicants respectfully disagree with this conclusion.

Paragraph 69 of Bjornson addresses the re-combining of subsequences into a complete sequence to report aggregate results of sequence analysis. The key referred to by Bjornson is just bookmarking to keep parent-child relationships intact. There is no equivalent disclosure of generating the consolidated key includes program code to decrease the depth value of the parent workload group such that the consolidated workload group is identified.

For at least these reasons, it is respectfully submitted that claim 33 is allowable.

Claim 35

Claim 35 recites, "The computer program product of claim 25, further comprising program code configured to construct a virtual key for mapping to the target resource, wherein the virtual key includes a hash value of the key identifier." In rejecting claim 35, the Office Action argues Bjornson discloses such limitations at Figure 5. OA, page 19. The Applicants respectfully disagree with this conclusion.

Figure 5 is relevant to our virtual key scheme as described previously. There is no equivalent disclosure of constructing a virtual key for mapping

to the target resource, wherein the virtual key includes a hash value of the key identifier.

For at least these reasons, it is respectfully submitted that claim 35 is allowable.

CONCLUSION

In view of the forgoing remarks, it is respectfully submitted that this case is now in condition for allowance and such action is respectfully requested. If any points remain at issue that the Examiner feels could best be resolved by a telephone interview, the Examiner is urged to contact the attorney below.

No fee is believed due with this Amendment, however, should a fee be required please charge Deposit Account 50-0510. Should any extensions of time be required, please consider this a petition thereof and charge Deposit Account 50-0510 the required fee.

Respectfully submitted,

Dated: January 14, 2008

/ido tuchman/

Ido Tuchman, Reg. No. 45,924
Law Office of Ido Tuchman
82-70 Beverly Road
Kew Gardens, NY 11415
Telephone (718) 544-1110
Facsimile (866) 607-8538